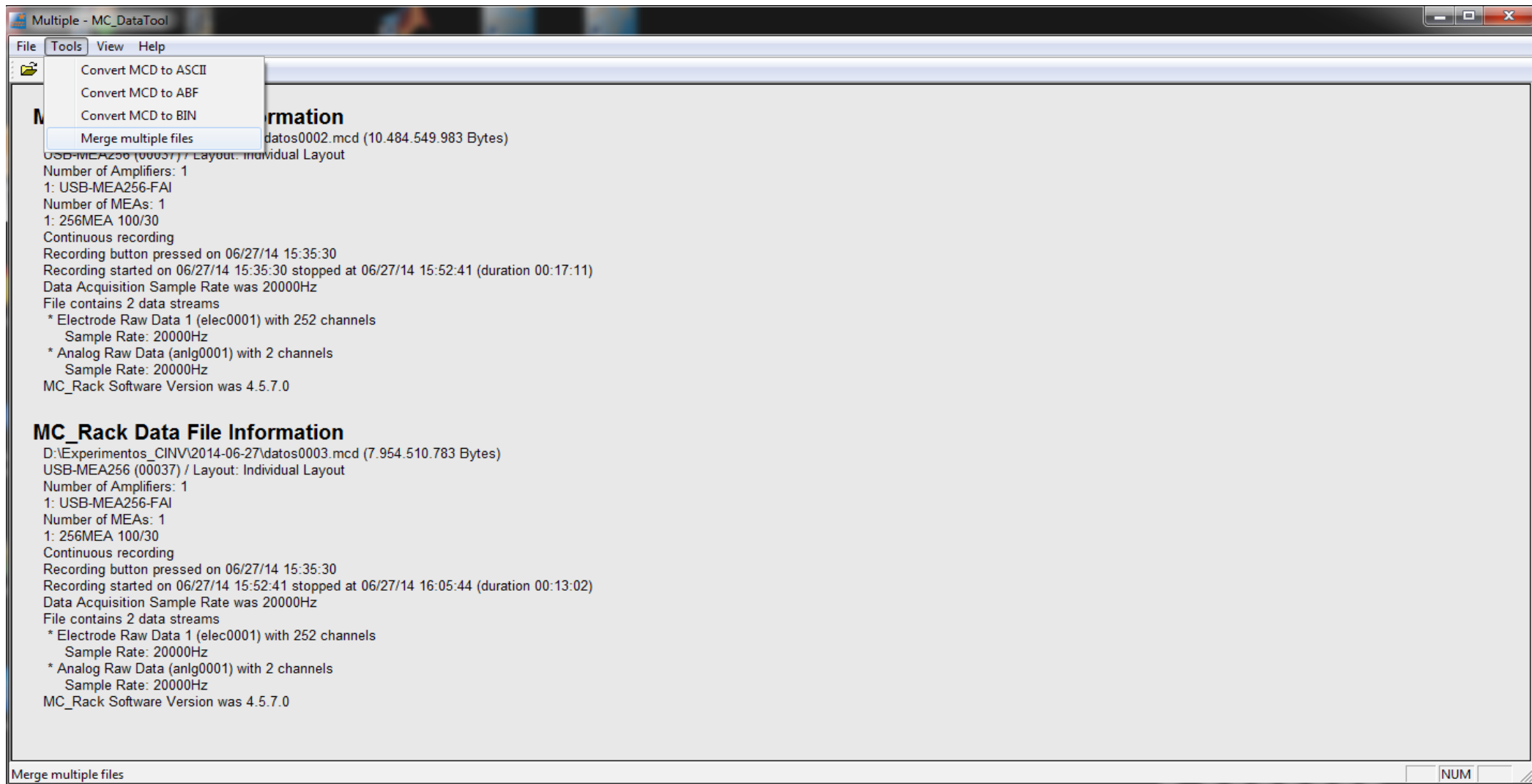


# Después del registro...

- .mcd
- Seed
- Bitácora
- Log File

# Si los .mcd están en varios archivos

## Mezclar .mcd a partir del MCD Data Tool

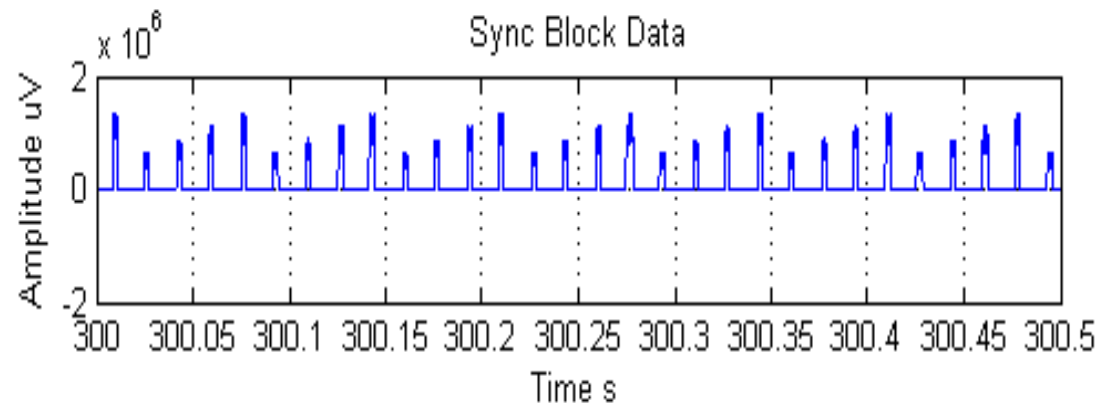


# Extraer y analizar señal de sincronía

Consta de 2 pasos:

1. `get_synchrony_signal ( .mcd x)`

este script da como salida un `.mat` con la señal de sincronía

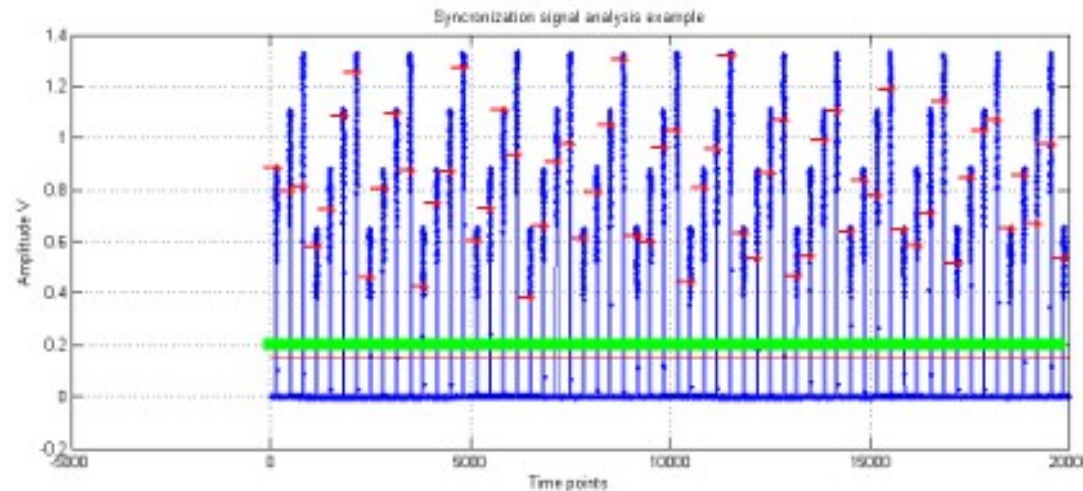


# Extraer y analizar señal de sincronía

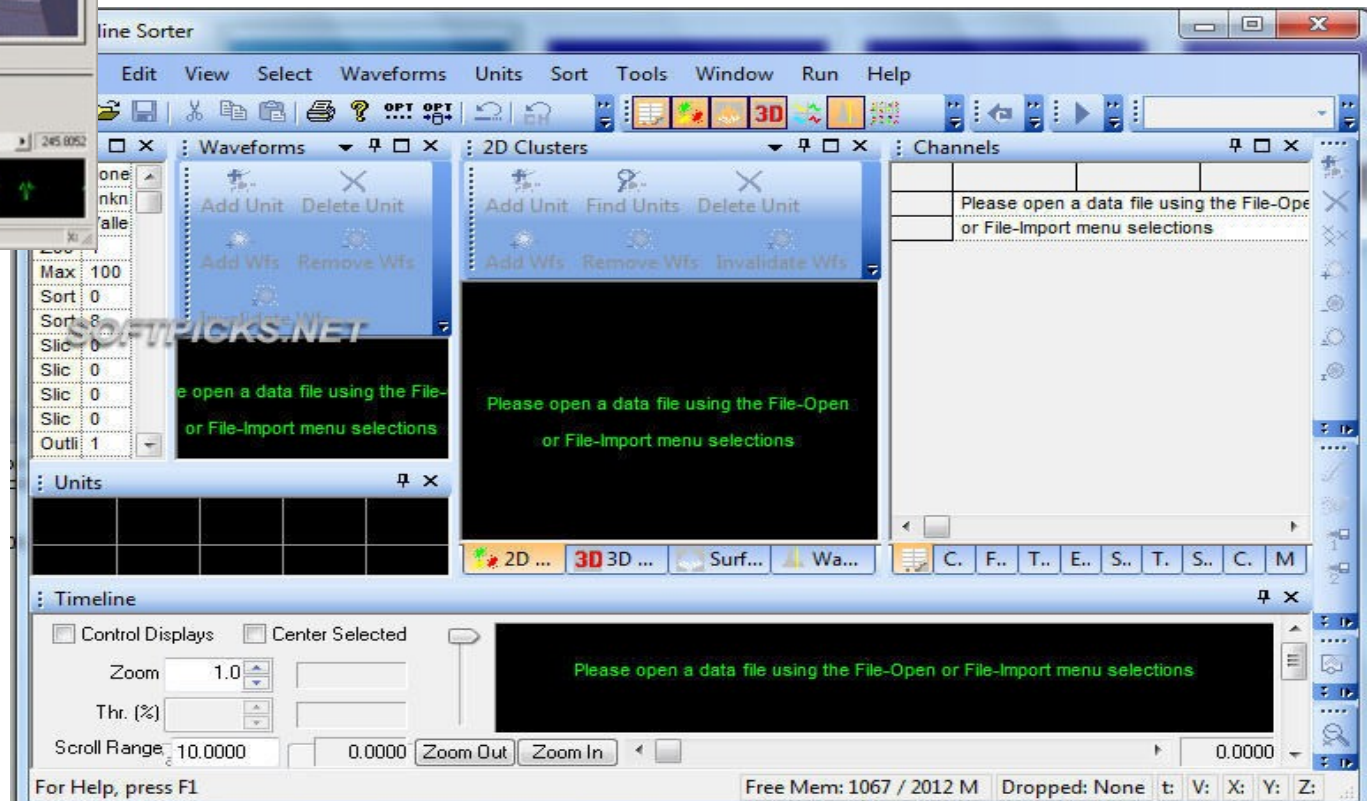
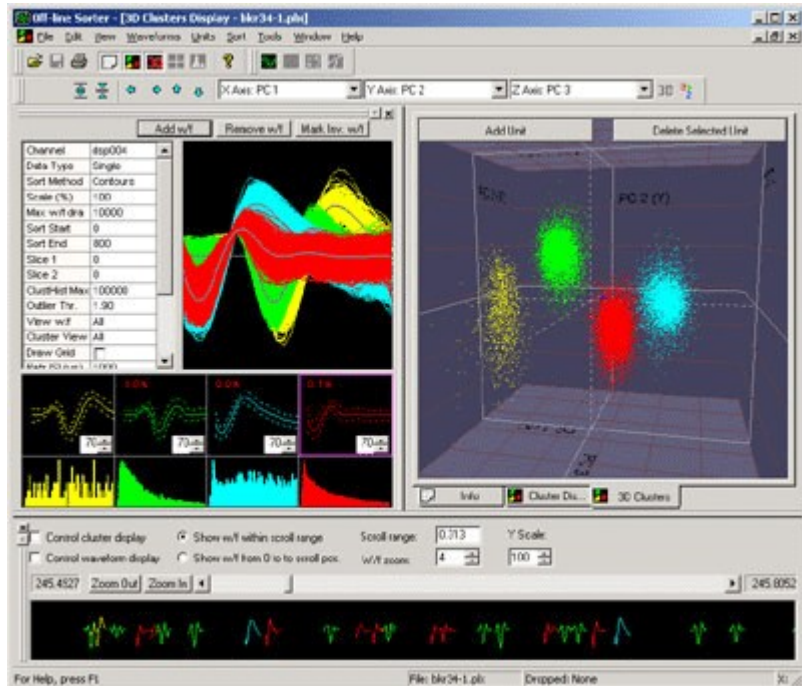
Consta de 2 pasos:

2. `syn_signal_analyzer` ( .mat)

este script da como salida un .txt con los datos de inicio y fin de cada frame del estímulo

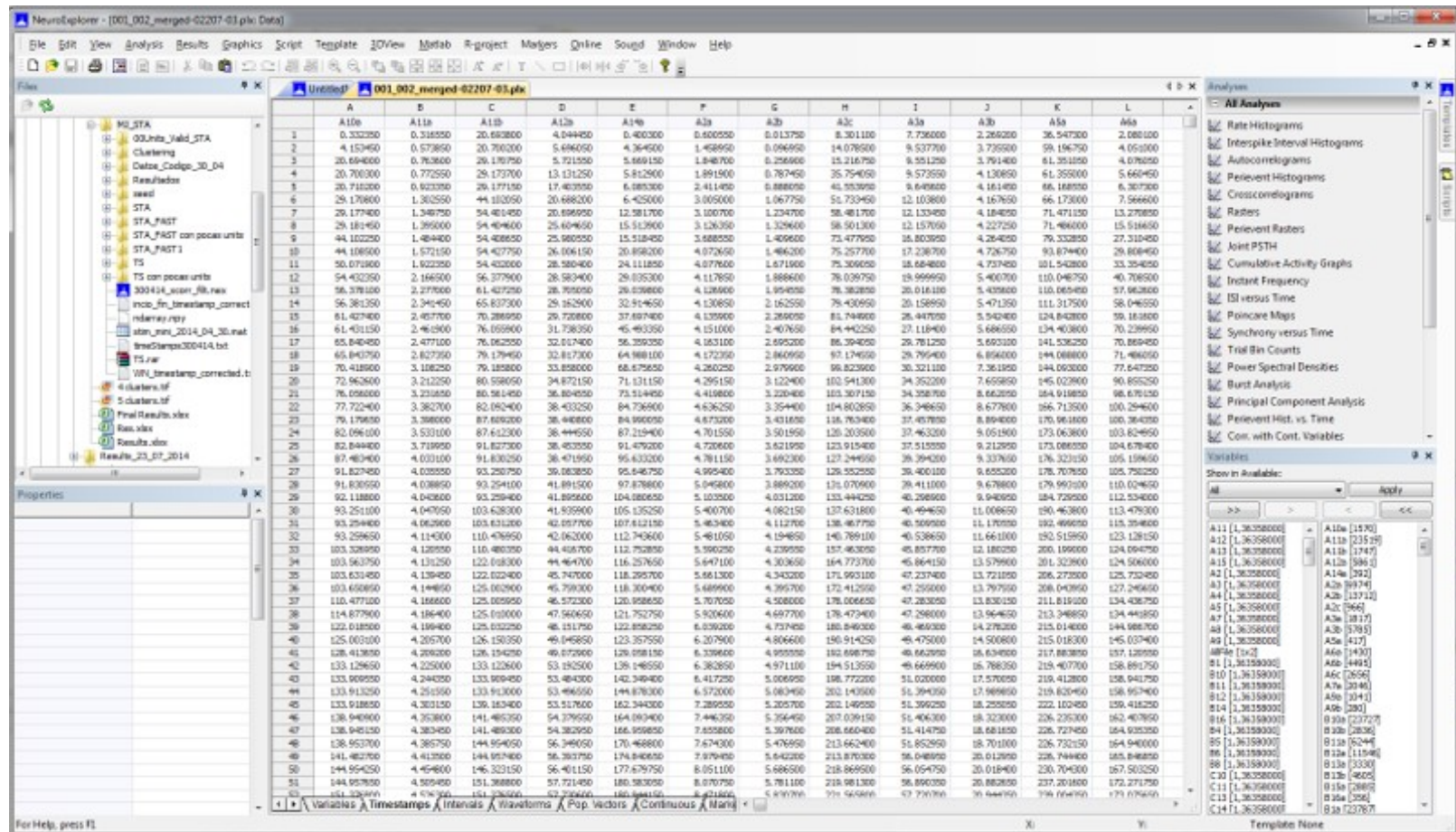


# Spike Sorting





# Neuroexplorer para guardar tiempos de spikes



# Scripts para generar parámetros del STA

## En python

- readwrite\_timestamps.py
  - - sourceFolder
  - - outputFolder

genera una carpeta que contiene todas las units y sus timestamps separados por subcarpetas

## En matlab

- `get_stim_from_seed_sta( seed_file,first_file,nimag,blocks, outputname ,do_v7 )`

genera un .mat con el estímulo presentado

# Mínimos parámetros para STA

Script en python sta.py

--sourceFolder (Fichero donde se encuentran los timestamps de cada unidad)

--startUnit

--endUnit

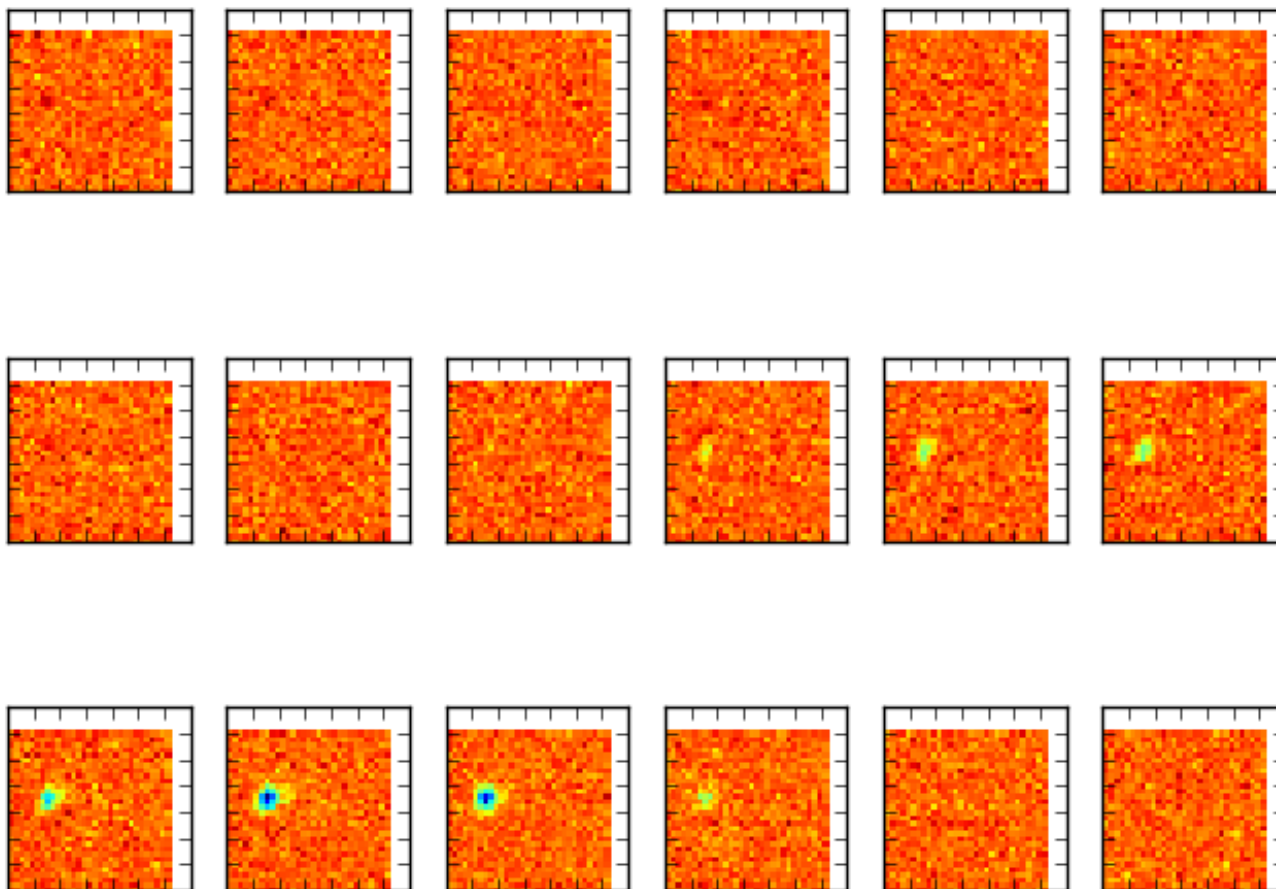
--outputFolder

--syncFile (fichero de inicio\_fin\_frames)

--stimMiniv7 o --stimMiniv73 (fichero del estímulo, para más de 100 000 imágenes tiene que usarse v73)



# STA



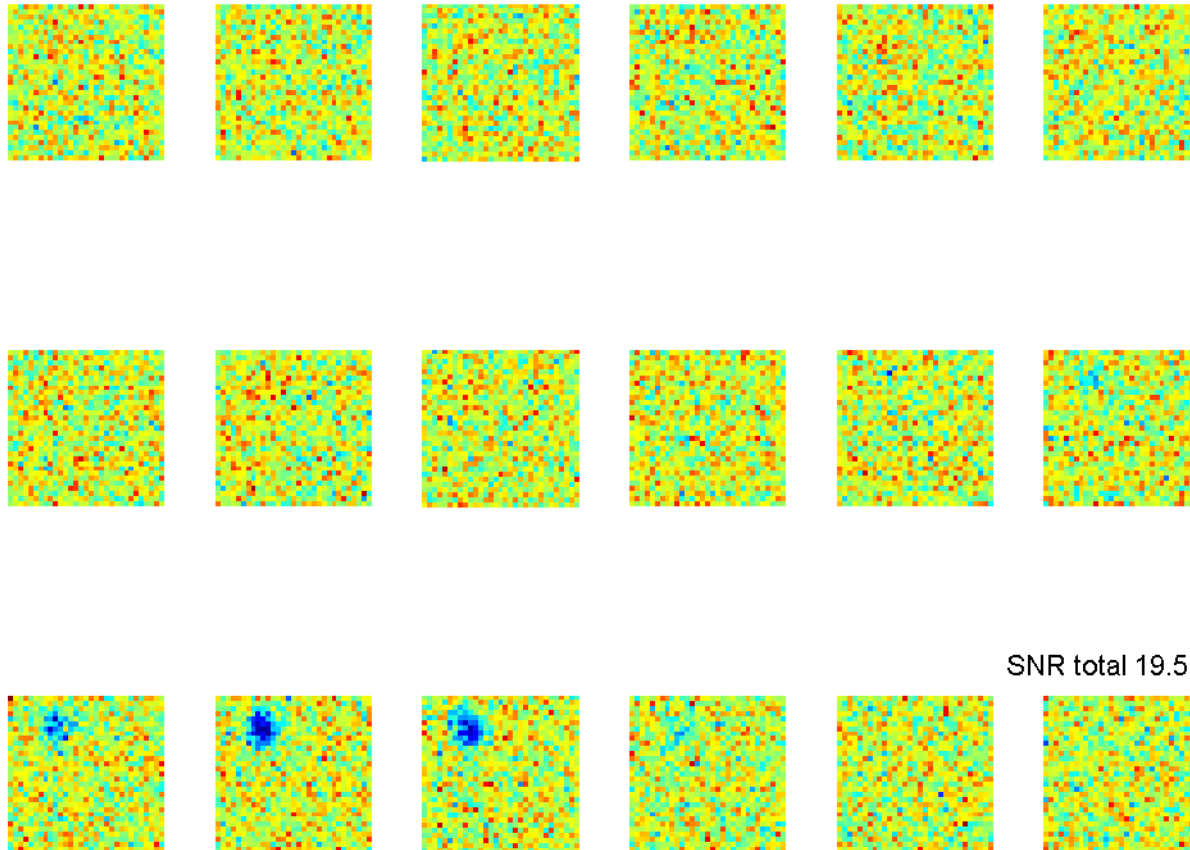
# Después del STA

Script en matlab para calcular calidad del STA:

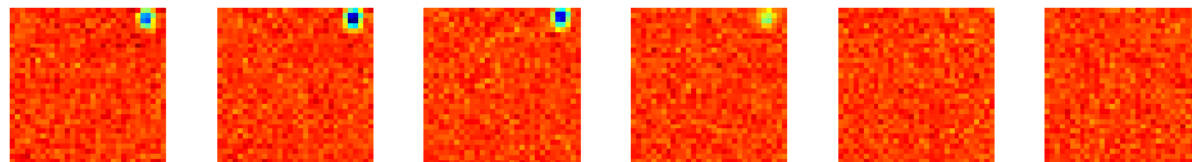
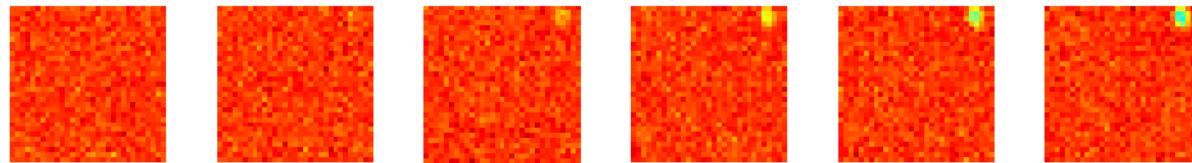
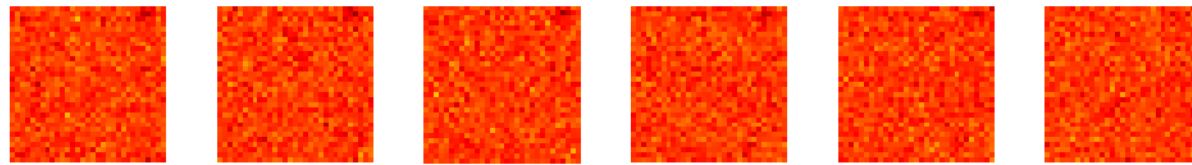
- `sta_inspector (units)` genera un fichero con cada unidad y su SNR y además imágenes con un valor de SNR calculado como:

```
snr_total=20*log10(abs(max_total-min_total)./ims_total)
```

# Imagen resultante del STA\_inspector



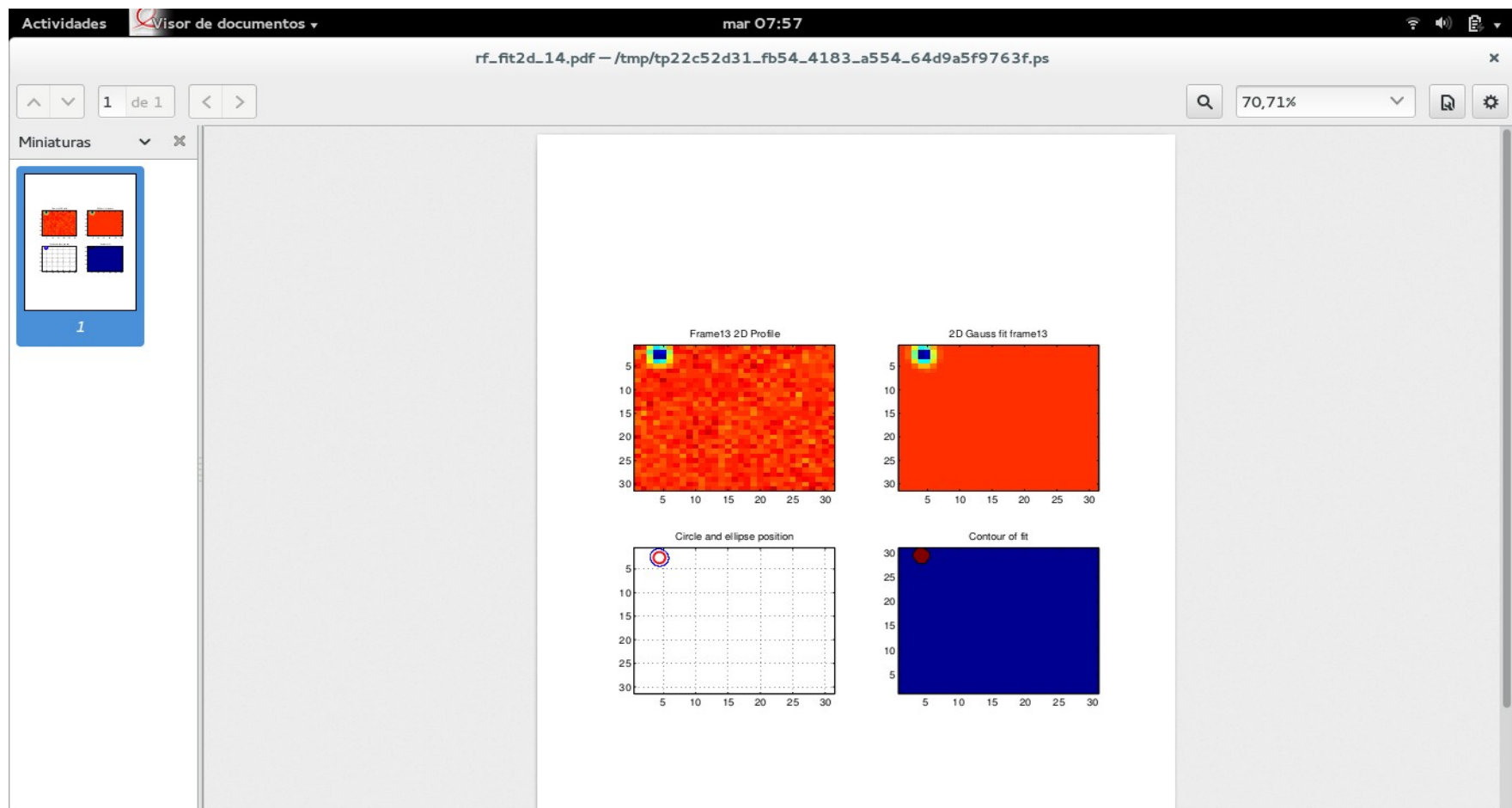
# Imagen resultante del STA\_inspector



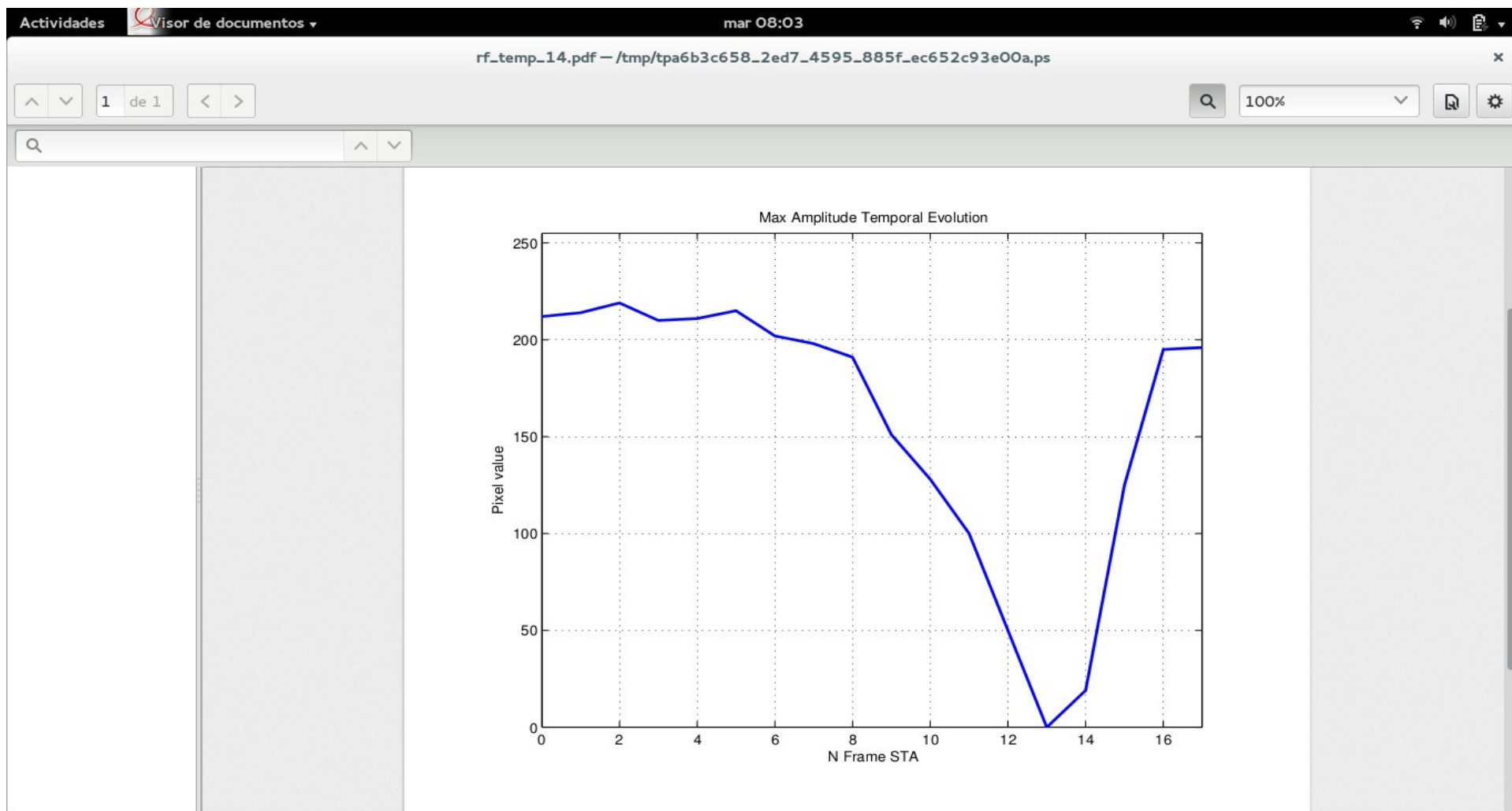
SNR total 29.1045

# Ajuste Gaussiano

- Script en matlab : Gauss2dfitSTA  
( unit\_STA.mat )



# Ajuste Gaussiano



# Clustering

- Realiza el método de clasificación Kmeans a los datos resultantes del STA despues de haberles realizado ajuste gaussiano.
- K-means es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de  $n$  observaciones en  $k$  grupos en el que cada observación pertenece al grupo más cercano a la media.

Script en python `pca.py`

`--sourceFolder` (default: None)

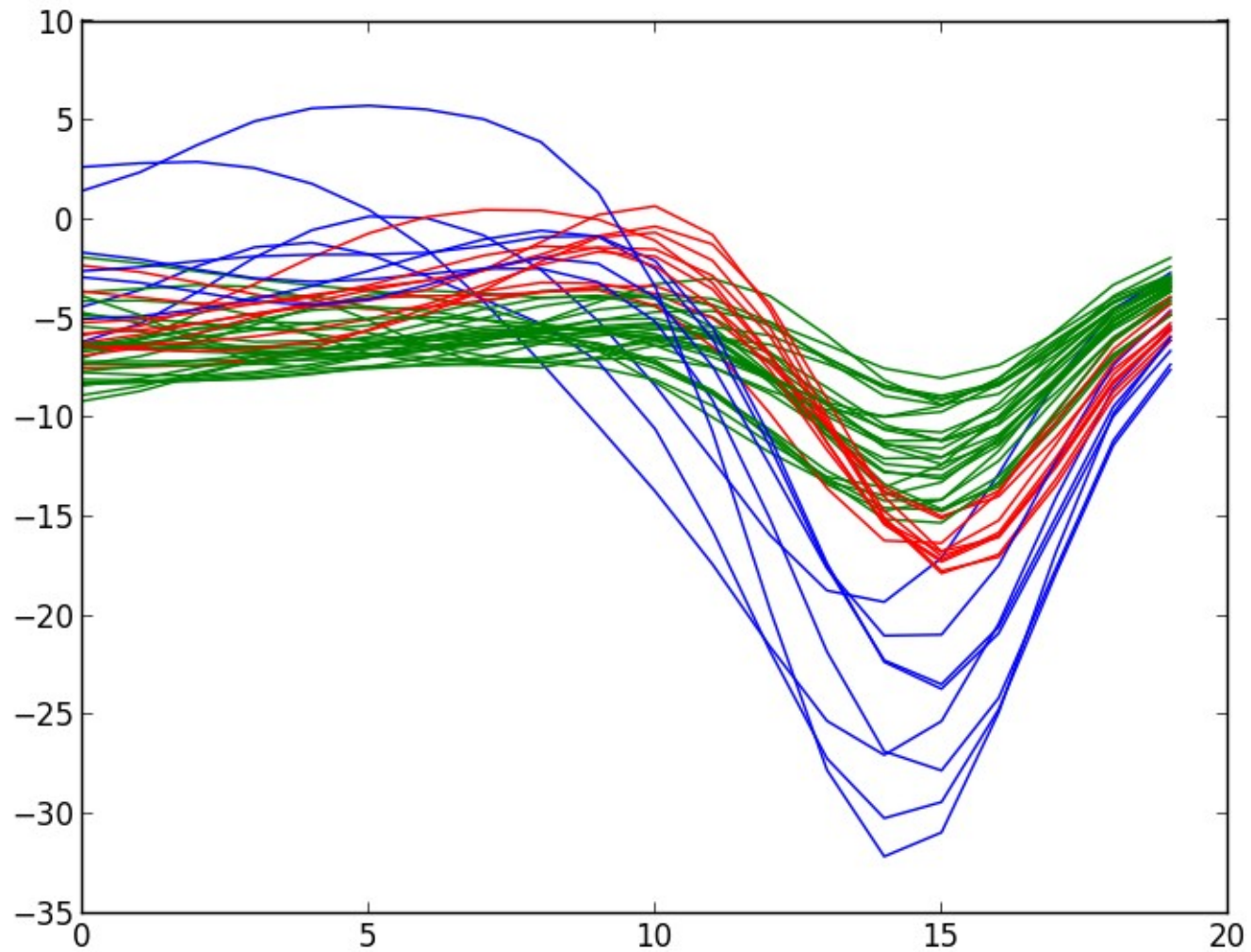
`--outputFolder` (default: None)

`--clustersNumber` (default: 5)

`--framesNumber` (default: 20)



# Clustering



# Tabla de resultados

Units	Spikes	STA	ON/OFF	Radio A	Radio B	Area	Angulo	Posición X	Posición Y	SNR
B15a	2885	1	Off	1.3006851201	1.03204113421	4.21501211629	62.8855988009	30.8227840862	1.00000103719	31,9353240254
B13b	4605	1	Off	1.17006650524	1.70512592043	6.26464768188	104.531913575	28.4194124153	1.00057344724	30,9525442713
A12a	5861	1	Off	1.66769705604	2.72214233579	14.2546855049	96.7873154936	27.4748328147	1.00000139291	30,9370728965
C15b	5429	1	Off	2.81631393368	1.67907029193	14.8483976447	80.49773605	30.9999999858	2.33007729672	30,4107386919
D15a	3413	1	Off	1.30785637347	1.98943912183	8.16996799423	14.4732821278	31.0	5.29253930946	29,9090269943